

Milad Mikeal

11/13/18

CS 162

Project 4

Design:

- Main
 - Declare variables
 - Create team1, team2, and loser queue
 - Seed rand
 - Create main menu & add options
 - Create character menu & add options
 - Create runAgain menu & add options
 - Create viewLosers menu & add options
 - Prompt main
 - If user selects play
 - Prompt user for number of fighter/team
 - For each team
 - Prompt characters
 - Name character
 - Add to back of respective teams queue
 - Set name
 - Until one team is empty
 - Set char dead flags to false
 - While both characters are alive
 - Attack with team 1's front player
 - Defend with team 2's front player
 - Check if team 2 char is dead
 - If not, attack with team 2's front player
 - Defend with team 1's front player
 - For winning team's front player
 - Increment winning team points
 - Move winning player to back of team's queue
 - Move losing player to back of loser queue
 - Count points and declare winning team
 - Prompt user if they want to see loser menu
 - Prompt user if they want to play again
- Queue class
 - Brought over from lab 7 with minor adjustments
- Character classes
 - Brought over from project 3

Main Menu Test Table: 1 – Start, 2 – Quit

Test Case	Input	Expected Outcome	Outcome
Input too low	Input < 1	Prompt user to re-enter value	User prompted to re-enter value
Input too high	Input > 2	Prompt user to re-enter value	User prompted to re-enter value
Input in range	Input = 1 or Input = 2	Proceed accordingly depending on user's input	Program proceeded accordingly
Non-numeric input	Input = alpha	Prompt user to re-enter value	User prompted to re-enter value

Characters Choice Menu:

Test Case	Input	Expected Outcome	Outcome
Input too low	Input < 1	Prompt user to re-enter value	User prompted to re-enter value
Input too high	Input > 5	Prompt user to re-enter value	User prompted to re-enter value
Input in range	Input between 1 - 5	Proceed accordingly depending on user's input	Program proceeded accordingly
Non-numeric input	Input = alpha	Prompt user to re-enter value	User prompted to re-enter value

View Losers Menu: 1 – Yes, 2 – No

Test Case	Input	Expected Outcome	Outcome
Input too low	Input < 1	Prompt user to re-enter value	User prompted to re-enter value
Input too high	Input > 2	Prompt user to re-enter value	User prompted to re-enter value
Input in range	Input = 1 or Input = 2	Proceed accordingly depending on user's input	Program proceeded accordingly
Non-numeric input	Input = alpha	Prompt user to re-enter value	User prompted to re-enter value

Play Again Menu: 1 – Yes, 2 – No

Test Case	Input	Expected Outcome	Outcome
Input too low	Input < 1	Prompt user to re-enter value	User prompted to re-enter value
Input too high	Input > 2	Prompt user to re-enter value	User prompted to re-enter value
Input in range	Input = 1 or Input = 2	Proceed accordingly depending on user's input	Program proceeded accordingly
Non-numeric input	Input = alpha	Prompt user to re-enter value	User prompted to re-enter value

Reflection:

I really enjoyed this assignment. I essentially ended up being a combination of project 3 and lab 7. I did not really find it difficult. I actually enjoyed piecing together all the pieces and getting a working, functional program.

I decided to use a circular linked list. I just brought it over from lab 7 and made a few minor adjustments. Most notably the addBack method. The only truly tricky part was adding back the appropriate character that the user selects from the menu in main. I decided that I would attribute each character with a respective number: 1 – Vampire, 2 – Barbarian, 3 – Blue Men, 4 – Medusa, 5 – Harry Potter. I added that attribute for each character class, then used the get method for that variable and passed it as the argument for addBack. Then in add back, I created the necessary character based on that number passed as an argument. Beyond that, nothing else was too tricky.

Once again, I thought this was a very fun assignment. I learned a lot my implementing a queue on actual objects as opposed to basic integers. The project gave me a good grip on chaining methods together with objects, for example team2.getFront()->getStrength().

